REMARKS

The Examiner is thanked for the comments in the Action. They have helped us considerably in understanding the Action and in drafting this Response thereto. It is our understanding that claims 1-11 remain pending in this application.

We proceed now with reference specifically to the numbered items in the Action.

Items 1-2 (§ 102(e) rejections):

Claims 1-11 (all) are rejected as being anticipated by Greene et al. Respectfully this is error.

With respect to claims 1 and 5, the Action states:

Greene et al discloses a search engine, comprising:

...

a hash pointer unit able to store a plurality of pointer values, wherein respective said pointer values are addressed based on said hash addresses (See column 4, lines 15-24, also see abstract); ...

However, the assertion here is not supported by the citation. The cite encompasses two paragraphs, respectively addressing when a key maps and when it does not. For the sake of simplicity we can consider just the first of these, since the second is straightforward once the first is grasped. The first (col. 4, lines 15-19) states:

> *According to another aspect of the above embodiment, when one input key maps to a first memory location at which there is no collision, the first memory location points to an entry in a third memory that includes a key value and associated data corresponding to a table key.*

The cite clearly states here that an "*input key*" is being mapped, i.e., is the input being used, and this cannot be reconciled with the assertion (which correctly recites the language of claim 1). In the claimed invention the "*hash address*" is the input being used by the hash pointer unit, and this hash address is provided by the hash function in the controller based on an "*input search value*" (see e.g., claim 1, ln. 2).

Thus, what has been misinterpreted here is that the "*input key*" of Greene is equivalent to applicant's "*input search value*" (to the extent that such characterization is valid at all), and the hash pointer unit of the claimed invention does not work with the input search value – it works with a hash address derived from the input search value by the hash function in the controller.

The advantages of the claimed invention operating in this novel manner are discussed in the specification, for example, in paragraphs [0044] – [0049].

As for the rest of this rejection, other assertions about elements of the claimed invention here are also not supported by their citations. For example, the Action also states:

> *a controller including a hash function able to receive an input search value and to create there from at least one hash address which is smaller in size than said input search value (See column 3, lines 4-29);*

However, while the cite here does discuss a hash function, for reasons similar to those already covered it is one that cannot be reconciled with the presently claimed invention. Similarly, the Action states:

> *a memory suitable for storing a database of search results, wherein respective said search results are addressed based on said pointer values (See column 7, lines 4-46);*

The cite here, however, recites a second hash function and the claimed invention does not include such (discussed in more detail, below). Continuing, the Action states:

> *a result bus operationally connecting said memory to said controller and able to communicate said search result from said memory to said controller, thereby permitting the search engine to function in a multi-way set-associative manner wherein the size of said memory is not a function of the degree of multi-way set-associativity (See column 3, lines 30-53, also see abstract).*

Here the first citation is merely a discussion in the Background Of The Invention section of Greene about deficiencies in the prior art – without even an assertion here that Greene remedies these, and particularly not an enabling teaching of elements relevant to the present claimed invention. The second citation (to the Abstract of Greene) clearly recites elements and their relationships that cannot be reconciled with the presently claimed invention. For example, the second hash function, again.

Accordingly, the cited reference cannot support the present rejection because it fails to teach or reasonably suggest at least one substantial element of the claimed invention.

As for claims 2-8, these are dependant on claims 1 and 5 and should be allowable for at least the reasons discussed above.

As for claim 9, the Action states:

> *Greene et al discloses ...*

(b) creating a plurality of hash addresses based on said hash value and respective offset values (See column 6, lines 16-51);

(c) retrieving pointer values from a pre-stored plurality of said pointer values based on said hash addresses (See column 5, lines 9-42);

(d) retrieving instances of the search results from the database based on said pointer values (See column 5, lines 63-67); and

However, there is a single problem with all of these assertions that becomes apparent by examining their citations in detail. Greene teaches a scheme using a "*second hash function*" (see e.g., col. 6, ln. 25 and 32 (first and second hash functions) and col. 5, ln. 38 and 66; in particular, see also, col. 5, ln. 43-54). The claimed invention does not use a second hash function, it uses a single hash function. This can be seen because it generates a "hash value," singular in the claims. A plurality of hash addresses are then created based on this and offset values (conceptually, each a form of base-plus-offset derived address). Applicant's plurality of hash addresses are not generated from a second function, as apparently assumed to arrive at the rationale underlying the rejection here.

Continuing, the Action also states:

(e) comparing said input search value and said stored search values in said instances of the search results retrieved (See column 3, lines 1-27, also see column 7, lines 8-34) in said step (d) to determine whether a respective hash collision has occurred, ...

Granted, the claimed invention, as well as Greene and considerable other prior art all determine whether a hash collision has occurred by comparing the input value with a part of a search result. What is useful to consider here however, are the citations in Greene, wherein it discusses what the prior art does to an input key value to retrieve a search result (col. 3, ln. 4-27) and what it does to its input value to retrieve its search result (col. 7, ln. 8-34).

At col. 3, ln. 4-27 Greene discusses how the prior art uses a binary search tree to resolve to a search result that is not the result of a hash collision. The claimed invention, however, does not use and would not be able to use a binary search tree for such. In the claimed invention, an input search value is used to generate a hash value (singular) (in claim 9, step (a)). From this and offset values a number of hash addresses are then created (i.e., multiple addresses based on the single hash value, by combining it with multiple offsets) (step (b)) – but this is not done using a second hash function. Then pre-store pointer values are retrieved based on the hash addresses (step (c)), and these are used to retrieve search results from the database of possible search results (step (d)). Next, the original input search value is compared to respective values in each

of the retrieved search results, with a match denoting that a hash collision has not occurred and thus that such a search result is usable (step (e)).

Lets consider an example using 4-way associativity. A single input search value results in the generation of a single hash value (regardless of the degree of associativity). From this four hash addresses are created by combining the hash value with four offsets (e.g., this can be as simple as concatenating the binary values "00", "01", "10", and "11" to the hash value; the specification discusses this very case in paragraphs [0044] – [0049]). These four hash addresses are then used to retrieve four pointer values (either the ultimate addresses in the database or values easily used as the basis for calculating the ultimate addresses). With these, four search results are retrieved from the database, each containing a search value and possibly other data as well. The original input search value is then compared with each of these four search values, with any match thus indicating that the respective search result is not an erroneous one due to a hash collision. Obviously, there is no binary tree search used or even usable in any of this.

Continuing, at col. 7, ln. 8-34 Greene discusses how in it "*The third store 114 is accessed whenever either the first store* [104] *or the second store* [110] *returns a leaf pointer.*" See also, FIG. 1 of Greene -- clearly showing its reliance on two hash functions and three data stores. Two things have been missed in applying Greene to the presently claimed invention. First, as noted, Greene teaches a scheme based on the use of two hash functions and the claimed invention only uses a single hash function. Second, Greene teaches a scheme based on the use of three data stores and the claimed invention only uses two. The claimed invention has a database (one data store) and a pre-stored plurality of pointer values, implicitly in a second data store. No third data store is recited or needed.

Accordingly, the cited reference cannot support the present rejection. Greene fails to teach or reasonably suggest the steps of the claimed invention because it clearly has steps involving two hashing functions and three data stores. Furthermore, the prior art discussed in Greene is not applicable here, because it relies on a binary tree search and the claimed invention does not.

As for claims 10-11, these are dependant on claim 9 and should be allowable for at least the reasons discussed above.

Item 3 (Conclusion):

This appears informational in nature and is understood to require no reply.

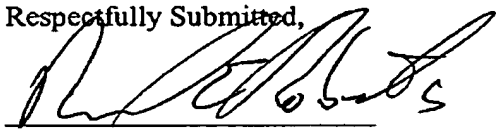
CONCLUSION

Applicant has endeavored to put this case into complete condition for allowance. It is thought that the §102 rejections are shown to be unfounded on the prior art reference cited. Applicant therefore asks that all objections and rejections now be withdrawn and that allowance of all claims presently in the case be granted.

Intellectual Property Law Offices
1901 S. Bascom Ave., Suite 660
Campbell, CA 95008

Telephone: 408.558.9950
Facsimile: 408.558.9960
E-mail: RRoberts@iplo.com

Respectfully Submitted,



Raymond E. Roberts
Reg. No.: 38,597

Customer No.:

32112